

Preface

This issue contains five papers and two Technical Notes on a variety of subjects: advances in software metrics, Component Broker support for JavaBeans**, dynamic tables in relational databases, software development performance, management of intellectual capital, Web service aggregation, and the thirtieth anniversary of IMS*.

As a result of interest in software measurement, software development organizations have been using a significant number of metrics and collecting substantial quantities of information about their processes, methods, tools, and results. As their methods, data quality, and needs change over time, these data collections and metrics should be re-examined. An organization can, for example, discover measurement goals that are not being met by the metrics or the data, goals that could be met by the collected data but where the data have not been exploited, and data that are collected and metrics that are in use but do not support organizational goals. Mendonça et al. combine top-down and bottom-up techniques to review, rationalize, improve, and rejuvenate the use of metrics and data within an organization.

Codella et al. have developed an understanding of IBM's Component Broker and Sun Microsystems' Enterprise JavaBeans** that shows how these two component models can support, augment, and enhance each other in their respective roles as part of object-oriented systems. The authors demonstrate the combined ability of these two middleware layers to provide an object environment that has such valued features as broad scalability, transactional support, and persistence. They also describe and argue for further extensions to this cooperative and complementary situation that will cement the relationship and provide needed functions for both middleware systems.

Recently developed and standardized additions to the Structured Query Language (SQL) environment provide users and programmers with powerful new

capabilities and generalized table concepts. As Fuh et al. point out in their paper, these include what they refer to collectively as dynamic tables, or run-time, explicitly defined derived tables: user-defined temporary tables, transition tables, user-defined table functions, and table locators. The authors have constructed an effective prototype in the context of IBM DATABASE 2* (DB2*) Common Server that demonstrates how to perform compile-time and run-time processing for these new tables, while creating the required linkages between the tables and their references.

Sawyer and Guinan have studied the production methods and social processes of 40 commercial software development teams. The authors measured and assessed the effects of those methods and processes on software quality and team performance. They found little support for methods or processes as the fundamental drivers of quality and performance, although the data show social processes as more important than production methods. But about 75 percent of the variability in quality and performance is not accounted for by these two factors. Suggestions and speculations are made about the missing factors.

Huang has authored an essay introducing a significant part of the knowledge management arena: the capture, exploitation, and management of intellectual capital assets. He describes IBM's program in this arena—the Intellectual Capital Management program—and provides its motivation in areas such as competitive responsiveness, team productivity, and core competencies. The author is particularly interested in the creation of, management of, and capitalization on collective knowledge across large organizations, such as IBM. The methods and tools for such capitalization have been and are being built and used today to provide competitive advantage to organizations, and they are described in this essay.

In a Technical Note, Zhao presents Web service aggregators and, in particular, IBM's WebEntree. Web

service aggregators serve the users as intermediaries that provide a single view of many diverse and distributed Web services. For example, a user can be provided with many services, select those of interest, use the services through a single interface, and have the aggregator perform such common, repetitious, and differing service functions as registration and authentication, without repeated user involvement.

This year marks the thirtieth year since the first time the message "IMS READY" was displayed on a customer's computer screen. Few software systems enjoy that degree of longevity and success, and few have had such far-reaching and long-lasting effects on business data processing. In his Technical Note, Blackman retraces those 30 years, describing the capabilities, innovations, and history of the IBM Information Management System*, better known as IMS, from the computer environments of the late 1960s to the Internet of the late 1990s. The author notes that today over 90 percent of Fortune 1000 companies use IMS as their database management system of choice.

The next issue of the *Journal* will be a special issue on Enterprise Solutions Structure.

Gene F. Hoffnagle
Editor

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Sun Microsystems, Inc.